CLAIMS:

What we claim as our invention is:

1. A processor for convolutional decoding, comprising:

   a register comprising a plurality of ordered bit positions; and

   update logic coupled to the register to receive a first signal indicative of a result of a first add-compare-select instruction and a second signal indicative of a result of a second add-compare-select instruction, and configured to update the contents of the register dependent upon the first and second signals; and

   wherein in the event the first and second signals are received substantially simultaneously, the update logic is configured to shift the contents of the register 2 bit positions in order thereby vacating 2 consecutive bit positions, to update one of the vacated bit positions dependent upon the first signal, and to update the other vacated bit position dependent upon the second signal.

2. The processor as recited in claim 1, wherein the first add-compare-select instruction and the second add-compare-select instruction each specify two add operations, a compare operation, and a select operation.

3. The processor as recited in claim 2, wherein the first add-compare-select instruction and the second add-compare-select instruction each specify a first pair of source operands and a second pair of source operands, and wherein each of the add operations specifies summing one of the first pair of source operands and one of the second pair of source operands.

4. The processor as recited in claim 2, wherein the compare operation comprises comparing results of the two add operations.

5. The processor as recited in claim 2, wherein the select operation comprises producing a result dependent upon a result of the compare operation.

6. The processor as recited in claim 2, wherein the processor is configured to execute the first add-compare-select instruction and the second add-compare-select instruction simultaneously dependent upon a set of instruction grouping rules.

7. The processor as recited in claim 6 further comprising an instruction sequencing unit and an execution unit, wherein the instruction sequencing unit is configured to issue decoded instructions to the execution unit for simultaneous execution dependent upon the set of instruction grouping rules.

8. The processor as recited in claim 1, wherein in the event the first signal is received and the second signal is not received, the update logic is configured to shift the contents of the register 1 bit position in order thereby vacating 1 bit position of the register, and to update the vacated bit position dependent upon the first signal.

9. The processor as recited in claim 1, wherein in the event the second signal is received and the first signal is not received, the update logic is configured to shift the contents of the register 1 bit position in order thereby vacating 1 bit position of the register, and to update the vacated bit position dependent upon the second signal.

10. A processor for decoding convolutional code, comprising:

an execution unit configured to produce a first signal and a second signal substantially simultaneously when executing two add-compare-select instructions at the same time, wherein the first and second signals are indicative of results of the two add-compare-select instructions;

a register comprising a plurality of ordered bit positions;

update logic coupled to the register and the execution unit to receive the first and second signals and to update the contents of the register dependent upon the first and second signals; and

wherein in the event the first and second signals are received substantially simultaneously, the update logic is configured to shift the contents of the register 2 bit positions in order thereby vacating 2 consecutive bit positions of the register, to update one of the vacated bit positions dependent upon the first signal, and to update the other vacated bit position dependent upon the second signal.

11. The processor as recited in claim 10, wherein each of the two add-compare-select instructions specifies two add operations, a compare operation, and a select operation.

12. The processor as recited in claim 11, wherein each of the two add-compare-select instructions specifies a first pair of source operands and a second pair of source operands, and wherein each of the two add operations specifies summing one of the first pair of source operands and one of the second pair of source operands.

13. The processor as recited in claim 11, wherein the compare operation comprises comparing results of the two add operations.

14. The processor as recited in claim 11, wherein the select operation comprises producing a result dependent upon a result of the compare operation.

15. The processor as recited in claim 10, wherein the two add-compare-select instructions are dispatched to the execution unit for simultaneous execution dependent upon a set of instruction grouping rules.

16. The processor as recited in claim 15 further comprising an instruction sequencing unit configured to issue decoded instructions to the execution unit for simultaneous execution dependent upon the set of instruction grouping rules.

17. A method for decoding convolutional code, comprising:

generating computer program code for a processor, wherein the computer program code comprises at least two add-compare-select instructions, and wherein storage elements specified by each of the at least two add-compare-select instructions are selected such that the processor will simultaneously execute the at least two add-compare-select instructions, and wherein the computer program code, when executed by the processor, causes the processor to carry out the following operations: (i) receiving symbols of the convolutional code in sequence, wherein each symbol has a corresponding input value used to generate the symbol, (ii) using the received symbols to build a data structure comprising data indicative of most likely transitions between nodes of a trellis diagram and input values associated with the most likely transitions, and (iii) using the data structure to produce the input values corresponding to the received symbols of the convolutional code; and

initiating execution of the computer program code by the processor.

18. The method as recited in claim 17, wherein the processor comprises a plurality of storage elements and is configured to simultaneously execute a plurality of add-compare-select instructions dependent upon a set of instruction grouping rules, and wherein each of the add-compare-select instructions specifies a subset of the storage elements, and wherein the grouping rules are directed to detecting storage element conflicts.

19. The method as recited in claim 18, wherein the storage elements specified by each of the at least two add-compare-select instructions are selected such that there are no storage element conflicts, the grouping rules are met, and the processor will simultaneously execute the at least two add-compare-select instructions.

20. The method as recited in claim 17, wherein using the received symbols to build the data structure comprises:

performing the following operations for each symbol of the convolutional code:

determining likelihoods of transitions between nodes of a trellis diagram;

selecting most likely transitions to nodes in a current stage of the trellis diagram; and

modifying a data structure to reflect the most likely transitions and input values associated with the most likely transitions.

21. A computer readable medium tangibly embodying program instructions operable to perform a method for decoding convolutional code, the method comprising:

receiving symbols of the convolutional code in sequence, wherein each symbol has a corresponding input value used to generate the symbol;

using the received symbols to build a data structure comprising data indicative of most likely transitions between nodes of a trellis diagram and input values associated with the most likely transitions;

using the data structure to produce the input values corresponding to the received symbols of the convolutional code; and

wherein the program instructions comprise at least two add-compare-select instructions specifying storage elements such that a processor executing the program instructions can simultaneously execute the at least two add-compare-select instructions.

22. The computer readable medium as recited in claim 21, wherein using the received symbols to build the data structure comprises:

performing the following operations for each symbol of the convolutional code:

determining likelihoods of transitions between nodes of a trellis diagram;

selecting most likely transitions to nodes in a current stage of the trellis diagram; and

modifying a data structure to reflect the most likely transitions and input values associated with the most likely transitions.